

Centro Universitário Newton Paiva

*BACHARELADO EM
SISTEMAS DE INFORMAÇÃO*

BANCO DE DADOS I

Prof. Iremar Nunes de Lima

iremar.prof@uol.com.br
<http://iremar.prof.sites.uol.com.br>

Proibida reprodução e distribuição desta apostila por quaisquer meio.

@Copyright By Iremar Nunes de Lima

Índice

1. INTRODUÇÃO	2
1.1 REVISÃO	3
1.2 FUNDAMENTOS DE BANCO DE DADOS	3
2. MODELO CONCEITUAL DOS DADOS.....	5
2.1 MODELO DE OBJETOS - DIAGRAMA DE CLASSES PERSISTENTES.....	5
2.3 ESTUDOS DE CASOS	11
3. MODELO RELACIONAL.....	13
3.1 MODELO RELACIONAL DE BANCO DE DADOS	13
3.2 MAPEAMENTO DO MODELO CONCEITUAL PARA O MODELO RELACIONAL.....	15
3.3 NORMALIZAÇÃO	20
4. A LINGUAGEM SQL - STRUCTURED QUERY LANGUAGE.....	21
4.1 DEFINIÇÃO E MANIPULAÇÃO DE DADOS EM SQL.....	22
4.2 CONSULTAS EM SQL.....	25
4.3 OPERADORES RELACIONAIS	26
5. EXERCÍCIOS.....	27
5.1 PRÁTICAS DA LINGUAGEM SQL EM LABORATÓRIO	27
5.2 ESTUDOS DE CASOS COMPLEMENTARES	30

1. Introdução

Bibliografia:

Livro Texto:

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005. 724 p.

Livros Complementares:

MACHADO, F.; ABREU, M. **Projeto de Banco de Dados: Uma Visão Prática**. 14. ed. São Paulo: Érica, 2006. 320 p.

TAYLOR, Allen G. **SQL para Dummies**. Rio de Janeiro: Campus, 2001. 409 p.

Conteúdo da Disciplina:

Parte I

- Conceitos fundamentais em banco de dados (capítulo 1 do livro texto).
 - ✓ Fundamentos de banco de dados.
 - ✓ Características do enfoque de banco de dados.
 - ✓ Usuários de banco de dados.
 - ✓ Vantagens de se utilizar um SGBD.
 - ✓ Implicações da abordagem de banco de dados.
- Conceitos e arquitetura do sistema de banco de dados (capítulo 2 do livro texto).
 - ✓ Modelos de Dados, Esquemas e Instâncias.
 - ✓ Arquitetura do SGBD e Independência de Dados.
 - ✓ Linguagens e Interfaces de Banco de Dados.
 - ✓ O ambiente do Sistema de Banco de Dados.
 - ✓ Classificação dos SGBDs.
- Modelo conceitual de banco de dados (capítulos 3 e 4 do livro texto)
 - ✓ Modelo de objetos (Diagrama de Classes Persistentes - UML).
 - ✓ Modelo E-R (DER) e Modelo E-R Estendido.
 - ✓ Estudos de Casos com utilização de software CASE.

Parte II

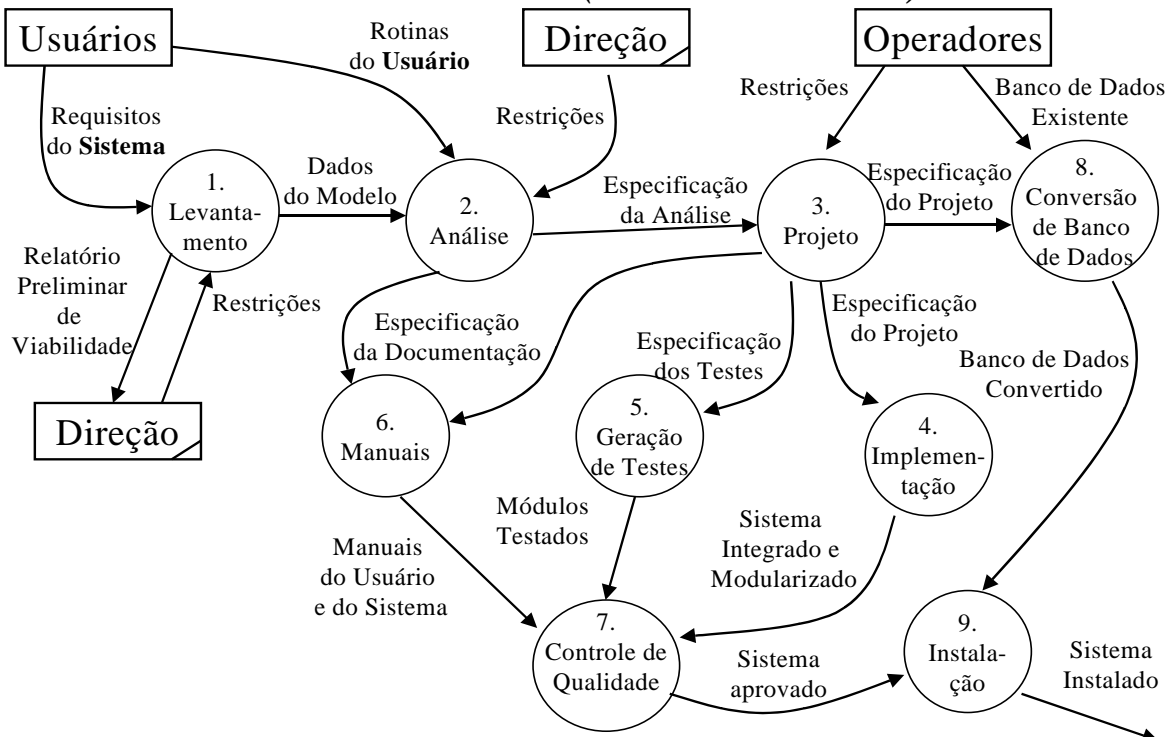
- Modelo relacional (capítulos 5, 7, 10 e 12 do livro texto)
 - ✓ Modelo relacional de dados.
 - ✓ Restrições Relacionais.
 - ✓ Mapeamento do modelo conceitual para o modelo relacional.
 - ✓ Normalização de bancos de dados relacionais.
 - ✓ Estudos de casos com utilização do software CASE Microsoft Visio.
- SQL (capítulos 8 e 9 do livro texto)
 - ✓ Definição de Dados, Restrições e Alterações de esquemas na SQL 92.
 - ✓ As Instruções Insert, Delete e Update em SQL.
 - ✓ Consultas básicas em SQL.
 - ✓ Inner Joins e Outer Joins em SQL.
 - ✓ Prática de SQL no SGBD Microsoft SQL Server.

1.1 REVISÃO

- **Sistema de Informação** é um conjunto de componentes inter-relacionados (programas) trabalhando juntos para coletar, recuperar, processar e armazenar informações com a finalidade de facilitar o planejamento e análise do processo decisório nas organizações.
- **Componentes lógicos de um SI:**



- **Ciclo de Vida do Desenvolvimento de SI (Método Estruturado):**



- Vamos supor o Levantamento de Dados já realizado através das diversas técnicas como Entrevistas, Brainstorming, Análise de documentação, JAD e Prototipação Descartável.

1.2 FUNDAMENTOS DE BANCO DE DADOS

- **Banco de Dados:** uma coleção de fatos conhecidos (dados) inter-relacionados.
- **Exemplos de Banco de Dados:**
 - ✓ Um catálogo de livros de uma biblioteca.
 - ✓ Uma agenda telefônica contendo nomes, endereços e telefones para contato.
 - ✓ Uma universidade contendo dados dos professores, alunos e disciplinas.
- Um banco de dados informatizado pode ser criado e mantido por um SGBD (Sistema de Gerência de Banco de Dados): um conjunto de programas que facilita o processo de definição, construção e manipulação de bancos de dados.
- Representação de um BD:
 - ✓ Pode ser representado na forma de tabelas.
 - ✓ Exemplo para um BD de um sistema de controle de pessoal de uma empresa:

Departamento	
Número	Nome
1	Pessoal
2	Vendas
3	Produção

Empregado				
Registro	Nome	Telefone	Salário	Departamento Lotado
212	José Antônio	33326262	1567,98	2
321	Antônio Rodrigues	32124589	345,84	3
109	Lucas da Silva	32124589	365,00	3
450	Maria Fernandes	33129860	2398,45	1
867	Carlos Henrique	35634011	689,00	3

Projeto			
Número	Nome	Data_Início	Data_Fim
001	Carro supersônico	02/03/2002	
002	Implantação ERP	23/01/2001	19/07/2001
003	Gestão de energia 2003	01/06/2002	

Empregado_Projeto		
NúmeroProjeto	RegistroEmpregado	HorasTrabalhadas
001	321	89
001	867	31
002	212	29
003	450	8
003	212	14

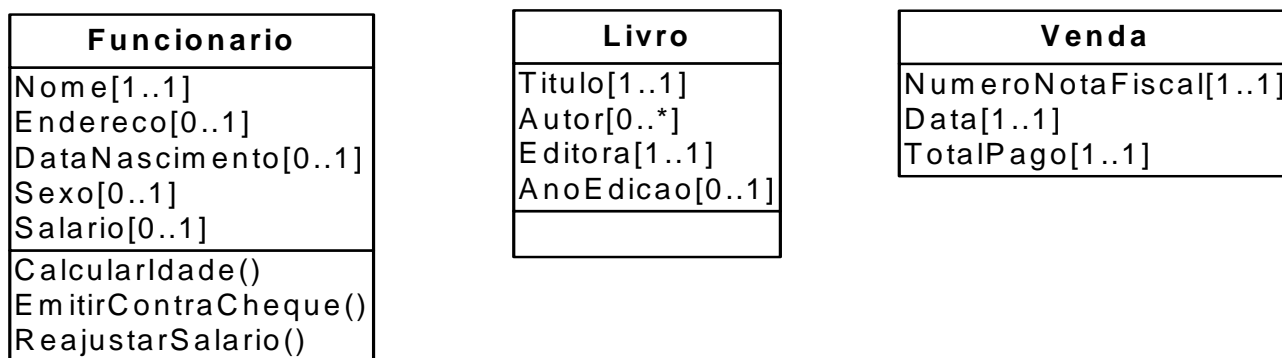
- *Melhor definição de um BD:* uma coleção de dados armazenados cujo conteúdo informativo a cada instante representa o estado de uma determinada aplicação de um grupo de usuários

2. Modelo Conceitual dos Dados

- Primeira etapa do projeto de Banco de Dados para um SI, cujo objetivo é obter uma *descrição abstrata dos dados independente do SGBD* a ser utilizado para a implementação.
- Baseado na percepção da realidade pelo analista após o levantamento de dados.
- Dois modelos se destacam atualmente:
 - ♦ Modelo de Objetos baseados na UML (Unified Modeling Language) – 1995.
 - ♦ Modelo ER (Entidade Relacionamento - DER) – 1976.

2.1 MODELO DE OBJETOS - DIAGRAMA DE CLASSES PERSISTENTES

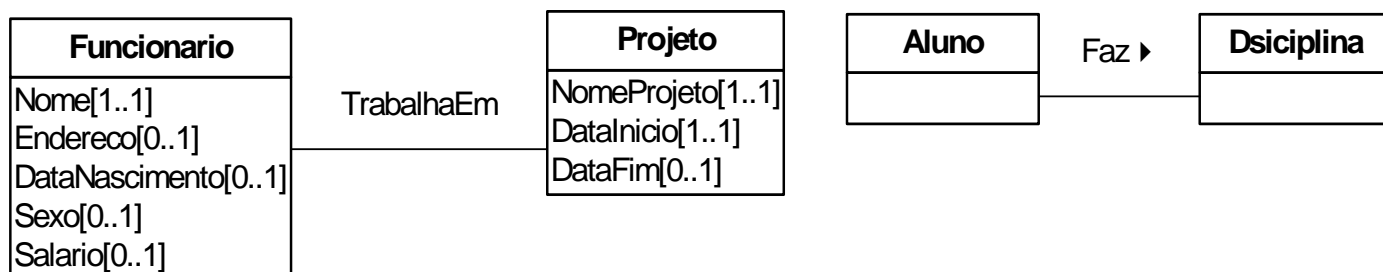
- Descreve os dados como objetos, classes, atributos, operações e relacionamentos.
- Classes: agrupamento de objetos similares que possuem atributos e operações semelhantes.
- Em banco de dados estaremos interessados nas *classes persistentes*, ou seja, nas classes que possuem objetos que precisam ficar armazenados no sistema por um período de tempo.
- Na UML uma classe é representada como nos exemplos a seguir:



- Nesta disciplina não indicaremos as operações das classes.
- Os atributos das classes podem ser opcionais ou obrigatórios, monovalorados ou multivalorados conforme mostrado nos exemplos acima.

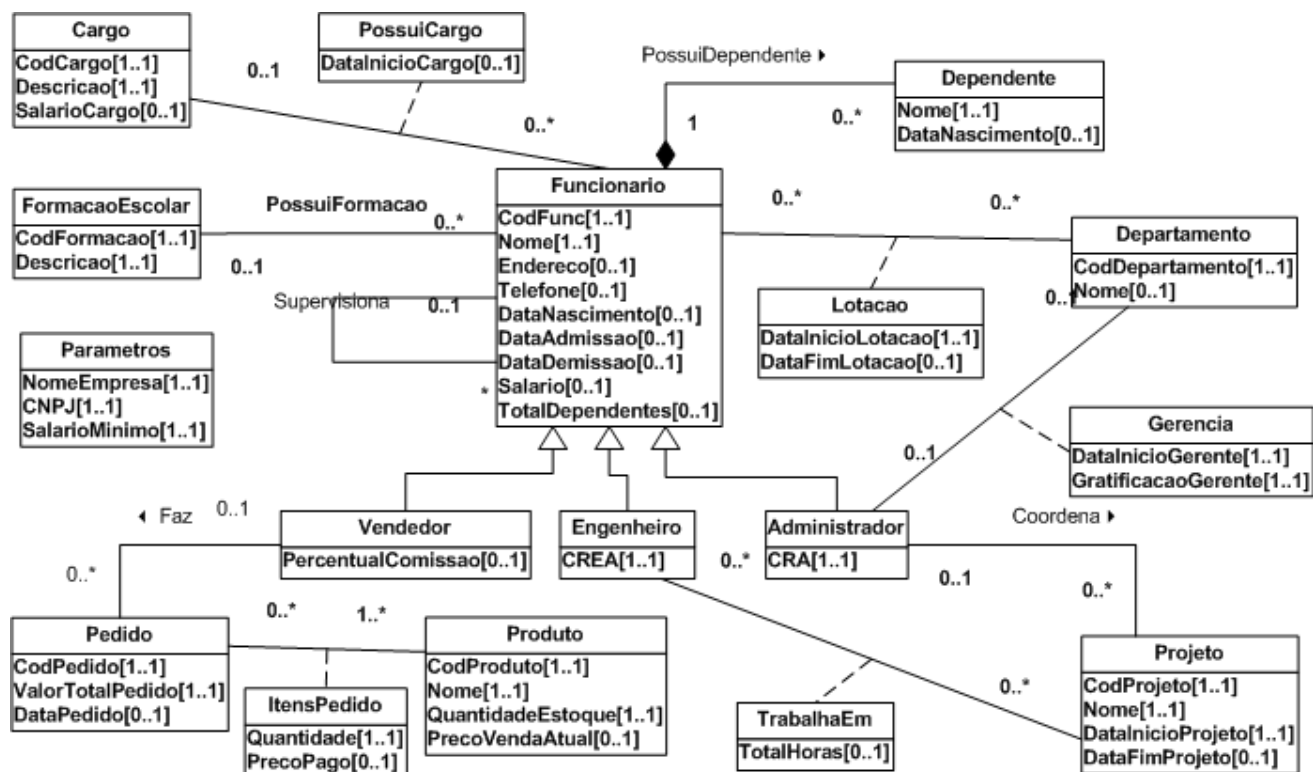
Associações entre Classes:

- Dependendo da regra de negócio, objetos de duas ou mais classes podem se associar indicando um novo **fato que será armazenado** no banco de dados.
- A representação em UML de uma associação entre duas classes é mostrada a seguir:



- *Nesta disciplina* só iremos criar associações entre classes se este fato tiver que ser armazenado no banco de dados por um período de tempo.
- Exemplos de associações em geral incorretas: Cliente *consulta* Produto, Funcionário *registra* Livro, Gerente *emite* relatório de Projetos.

Exemplo final: Diagrama de Classes para um sistema de controle de pessoal.



Observações:

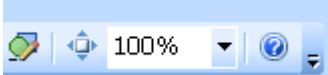
1. O atributo TotalDependentes é um atributo derivado.
2. O relacionamento Lotação guarda os históricos dos departamentos pelos quais os funcionários já ficaram lotados.
3. Não se está armazenando no Banco de Dados os históricos de Cargo dos funcionários e nem os históricos dos dias que cada funcionário trabalhou em um projeto.
4. Só se está armazenando o gerente atual do departamento. O mesmo vale para o coordenador atual do projeto.
5. As classes Engenheiro e Administrador foram criadas para melhor representar a regra de negócio da empresa. Podem também existir como objetos da classe cargo.
6. Ao demitir um funcionário seus dados não serão removidos do banco de dados. Observe a existência do atributo DataDemissão.
7. O relacionamento Supervisiona não pode ter participação total no lado um do relacionamento.
8. As classes Parametros e Usuario não se relacionam com nenhuma das outras classes.
9. Poderia existir uma classe GrupoUsuario para armazenar o perfil de segurança dos usuários do sistema. Isto é muito comum em sistemas de informação baseados em banco de dados.
10. A maioria das classes possui um atributo identificador (que será a chave primária da tabela que vai ser gerada no SGBD relacional).
11. Lembre-se: Este é um modelo didático e não pretende ser completo nem correto para um sistema de controle de pessoal de uma empresa real.

Como Configurar o Software Visio Para Desenhar o Diagrama Conceitual de Banco de Dados (segundo o padrão UML)

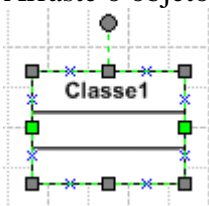
Objetivo: Aprender a usar e configurar o software Visio para construir o DCP - Diagrama de Classes Persistentes (Modelo Conceitual de Banco de Dados).

Siga as instruções abaixo:

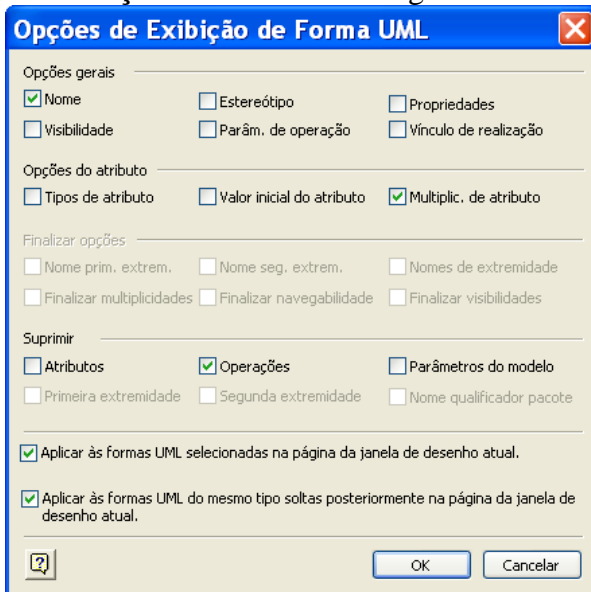
1. Inicie o Software Microsoft Visio.
2. Selecione a opção “Software e Banco de Dados” (Canto inferior esquerdo).
3. Selecione a opção “Diagrama de Modelo UML”.
4. Aumente o campo de visualização do ZOOM para no mínimo 80% (canto superior direito da sua tela conforme tela abaixo).



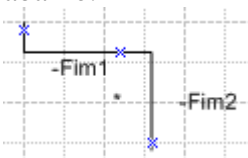
5. Arraste o objeto **Classe** (canto esquerdo da tela) para o centro da tela como mostrado abaixo.



6. Clique com o botão da direita do seu mouse em cima do desenho desta classe e selecione a opção “**Opções de Exibição de Forma**”. Configure conforme tela a seguir:



7. Desenhe uma associação arrastando o objeto **Associação Binária** (canto esquerdo da tela) como mostrado abaixo.



8. Clique com o botão da direita do seu mouse em cima do desenho da associação que foi desenhada anteriormente e selecione a opção “**Opções de Exibição de Forma**”. Configure conforme tela a seguir:

Opções de Exibição de Forma UML

Opções gerais

☒ Nome
 ☐ Estereótipo
 ☐ Propriedades
☐ Visibilidade
 ☐ Parâm. de operação
 ☐ Vínculo de realização

Opções do atributo

☐ Tipos de atributo
 ☐ Valor inicial do atributo
 ☐ Multiplic. de atributo

Finalizar opções

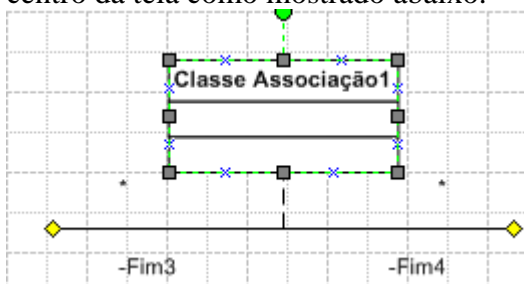
☐ Nome prim. extrem.
 ☐ Nome seg. extrem.
 ☐ Nomes de extremidade
☒ Finalizar multiplicidades
 ☐ Finalizar navegabilidade
 ☐ Finalizar visibilidades

Suprimir

☐ Atributos
 ☐ Operações
 ☐ Parâmetros do modelo
☐ Primeira extremidade
 ☐ Segunda extremidade
 ☐ Nome qualificador pacote

☒ Aplicar às formas UML selecionadas na página da janela de desenho atual.
☒ Aplicar às formas UML do mesmo tipo soltas posteriormente na página da janela de desenho atual.

9. Desenhe uma classe associativa arrastando o objeto **Classe Associação** (canto esquerdo da tela) para o centro da tela como mostrado abaixo.



10. Clique com o botão da direita do seu mouse em cima do desenho da Classe de associação que foi desenhada anteriormente e selecione a opção “**Opções de Exibição de Forma**”. Configure conforme tela a seguir:

Opções de Exibição de Forma UML

Opções gerais

☒ Nome ☐ Estereótipo ☐ Propriedades
☐ Visibilidade ☐ Parâm. de operação ☐ Vínculo de realização

Opções do atributo

☐ Tipos de atributo ☐ Valor inicial do atributo ☒ Multiplic. de atributo

Finalizar opções


☐ Nome prim. extrem. ☐ Nome seg. extrem. ☐ Nomes de extremidade
☒ Finalizar multiplicidades ☐ Finalizar navegabilidade ☐ Finalizar visibilidades

Suprimir

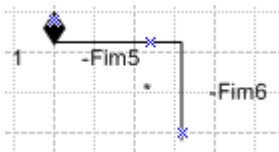
☐ Atributos ☒ Operações ☐ Parâmetros do modelo
☐ Primeira extremidade ☐ Segunda extremidade ☐ Nome qualificador pacote

☒ Aplicar às formas UML selecionadas na página da janela de desenho atual.


☒ Aplicar às formas UML do mesmo tipo soltas posteriormente na página da janela de desenho atual.



11. Desenhe um relacionamento de composição arrastando o objeto **Composition** (canto esquerdo da tela) para o centro da tela como mostrado abaixo.



12. Clique com o botão da direita do seu mouse em cima do desenho da Composição que foi desenhada anteriormente e selecione a opção “**Opções de Exibição de Forma**”. Configure conforme tela a seguir:

Opções de Exibição de Forma UML 

Opções gerais

<input checked="" type="checkbox"/> Nome	<input type="checkbox"/> Estereótipo	<input type="checkbox"/> Propriedades
<input type="checkbox"/> Visibilidade	<input type="checkbox"/> Parâm. de operação	<input type="checkbox"/> Vínculo de realização

Opções do atributo

<input type="checkbox"/> Tipos de atributo	<input type="checkbox"/> Valor inicial do atributo	<input type="checkbox"/> Multiplic. de atributo
--	--	---

Finalizar opções


<input type="checkbox"/> Nome prim. extrem.	<input type="checkbox"/> Nome seg. extrem.	<input type="checkbox"/> Nomes de extremidade
<input checked="" type="checkbox"/> Finalizar multiplicidades	<input type="checkbox"/> Finalizar navegabilidade	<input type="checkbox"/> Finalizar visibilidades

Suprimir

<input type="checkbox"/> Atributos	<input type="checkbox"/> Operações	<input type="checkbox"/> Parâmetros do modelo
<input type="checkbox"/> Primeira extremidade	<input type="checkbox"/> Segunda extremidade	<input type="checkbox"/> Nome qualificador pacote

☒ Aplicar às formas UML selecionadas na página da janela de desenho atual.

☒ Aplicar às formas UML do mesmo tipo soltas posteriormente na página da janela de desenho atual.



13. Clique em OK.

14. Agora que o software está configurado adequadamente, você deve *desenhar o DCP solicitado pelo professor*.

2.3 ESTUDOS DE CASOS

1. *Sistema para controle de biblioteca*

Considere a seguinte lista de eventos para o sistema de controle de uma biblioteca:

1. Atendente solicita registro de livro.
2. Atendente solicita registro de exemplar de livro.
3. Atendente solicita listagem de exemplares de livros da biblioteca.
4. Aluno solicita seu registro na biblioteca.
5. Atendente faz pedido de listagem de alunos cadastrados.
6. Aluno consulta livro por assunto.
7. Aluno consulta livro por autor.
8. Aluno solicita empréstimo de exemplar de livro.
9. Atendente solicita situação de empréstimo.
10. Aluno solicita devolução de exemplar de livro emprestado.
11. Bibliotecária solicita listagem de empréstimos vencidos.
12. Bibliotecária solicita disquete contendo dados de alunos em débito.

2. *Sistema para controle de uma livraria*

Considere a seguinte lista de eventos para o sistema de controle de uma livraria:

1. Diretor registra dados da empresa.
2. Diretor registra nova filial da empresa.
3. Diretor contrata vendedor para atuar em filial.
4. Vendedor registra livro (título, autor, editora, assunto).
5. Vendedor registra exemplares de livros adquiridos pela filial.
6. Cliente faz pedido de compra de exemplares de livro a ser tirado pelo vendedor.
7. Diretor solicita relatório contendo o total de vendas do dia ordenado por filial e forma de pagamento.
8. Diretor solicita listagem contendo o total de livros com estoque crítico por filial.
9. Diretor solicita emissão de contra cheque de vendedores com as respectivas comissões.

3. *Sistema para venda e locação de CDs*

Uma loja possui CDs para venda e locação. De acordo com o usuário um CD possui um título, preço de venda, data de lançamento, quantidade em estoque para venda e para locação. Um CD pode ter vários exemplares representando o CD físico que vai ser locado ou vendido, possuindo o número do exemplar, preço de compra e data da aquisição. Um exemplar do CD só pode ser vendido se ele for novo (nunca foi locado). Um cliente pode comprar ou locar uma quantidade ilimitada de exemplares de CDs. Para locar é obrigatório que o cliente esteja cadastrado na loja indicando o nome, endereço, CPF, data de nascimento e telefone. As vendas de CDs podem ser pagas de diversas formas como à vista com 10% de desconto, através de cheque pré-datado (30 dias), cartão de crédito, etc. As locações somente podem ser pagas à vista, no ato da devolução dos CDs, que tem de acontecer 4 dias após a locação. Se houver atraso é cobrado uma multa. A loja possui alguns funcionários cuja função é atender os clientes durante a venda e locação dos CDs. Suas principais tarefas são: atender novas locações, devoluções, receber pagamentos, emitir recibos de venda e locação pelo sistema, conferir manualmente o estado dos CDs devolvidos (caixa, disco e encarte), cancelar uma venda, etc. Os funcionários ganham além do salário fixo uma comissão de 2% sobre o total de vendas de CDs realizadas ao longo do mês. Nenhum controle de fornecedores (compras de CDs) vai ser feito neste sistema. Os usuários indicaram, entre outros, os seguintes relatórios a serem emitidos pelo sistema: Clientes em débitos com locações, Mala direta para os clientes aniversariantes, CDs com estoque crítico (abaixo de 10 exemplares para venda), os 10 CDs mais vendidos na semana, o fluxo de caixa contendo a movimentação de locações e vendas do dia com os respectivos valores recebidos, nota fiscal automatizada para as vendas de CDs (esta nota pode ser emitida a qualquer momento no sistema), folha de pagamento dos funcionários ao final do mês para controle da gerência, contra cheques dos funcionários e total das vendas do dia classificados por forma de pagamento.

4. *Sistema para controle de uma faculdade*

Considere a seguinte lista de eventos para o sistema de controle de uma faculdade:

1. Faculdade registra grade curricular (disciplina e pré requisitos) de curso.
2. Secretaria registra aluno aprovado no vestibular para um determinado curso.
3. Coordenador define turma de disciplina para o novo semestre.
4. Aluno solicita matrícula em turma de disciplina.
5. Coordenador contrata professor para lecionar turma de disciplina.
6. Professor lança frequência mensal de aluno na turma de disciplina que leciona.
7. Professor lança nota de aluno na turma de disciplina que leciona.
8. Aluno consulta boletim escolar contendo notas e frequências.
9. Aluno solicita trancamento de matrícula em uma dada turma de disciplina.

5. Sistema disque pizza

O sistema de controle para uma pizzaria tem como finalidade dar suporte ao controle de cadastros de clientes, produtos (pizzas semi-prontas, refrigerantes, cervejas, etc.), fornecedores, pedidos de venda aos clientes, pedidos de compra aos fornecedores, contas a pagar e a receber. Ao realizar o pedido o cliente informa o seu telefone e o sistema identificará se o cliente está cadastrado (nome, endereço e telefone). Depois de confirmado o pedido pelo funcionário o cliente indica a forma de pagamento e fica aguardando a entrega. O sistema deve emitir uma ordem de entrega, dar baixa no estoque para que se possa emitir pedidos de compras aos fornecedores, se necessário. O sistema possuirá todo um cadastro de fornecedores com seus respectivos produtos (cotação de preços) para auxiliar na compra dos mesmos. O sistema emitirá relatórios diários de fluxo de caixa (compras e vendas) para a gerência. Também deve ser emitida a relação de produtos com estoque crítico, consultas sobre a situação de um pedido de um cliente ou de um fornecedor (em aberto, cancelado, a caminho, finalizado, pago, etc), mala direta, relação de pedidos a vencer dos fornecedores e relação do melhor fornecedor para um dado produto com estoque crítico. Nenhum controle de funcionários deve ser feito neste sistema.

6. Sistema para uma vídeo locadora

Uma vídeo locadora possui várias filiais espalhadas pelo Brasil. Um cliente está vinculado a uma das filiais e pode ser um cliente titular ou um cliente dependente. A vídeo locadora possui várias obras e cada obra pode corresponder a diversos tipos como filme, peça de teatro, concerto musical entre outros. Uma obra pode ter diversos exemplares adquiridos em datas diferentes e com preços de compras diferentes. Uma obra sempre é classificada em diferentes categorias como lançamento, ouro, prata, entre outros, o que define o preço base da locação. Um obra também recebe uma única classificação de gênero como comédia, romance, musical, terror, etc o que define a impropriedade (censura) da obra. Um dado exemplar só pode ser feito de uma determinada mídia como DVD, Fita, Cartucho (de jogos), etc. Dependendo da mídia pode ser acrescido um determinado valor no caso de locação. Um cliente pode locar quantos exemplares quiser e se desejar pode também comprar um ou mais exemplares. Neste caso é importante manter o preço de venda da obra, a data de venda e o cliente que comprou para emissão de relatórios semanais de vendas e mala direta com promoções para clientes. A locadora possibilita a reserva de uma obra caso nenhum dos exemplares estejam disponíveis. É possível realizar diversas consultas como saber quais os atores e diretores de um filme e o valor mensal arrecadado com locações, clientes com locação em débito, valor total arrecado com multas no final do mês, relação semanal das 10 obras mais locadas, gênero mais procurado, malas diretas para clientes aniversariantes. Nenhum controle de funcionários e fornecedores (compras) deve ser feito pelo sistema. Para efeitos de controle do usuário é importante que seja implementado as seguintes rotinas: cancelar uma locação, reserva ou venda, emissão de nota fiscal de venda a qualquer momento, renovar locação, totalização diária de locação e vendas por filiais e relação diária de obras disponíveis classificadas por mídia para clientes que as tenham reservadas.

7. Sistema para uma clínica médica

Uma clínica médica possui vários médicos atuantes em diversas especialidades como clínico geral, cardiologia, pediatria, oftalmologia, etc. Um médico pode atuar em diferentes especialidades e pode atender em diversos horários do dia e da semana. Um paciente só pode marcar uma consulta se estiver cadastrado. Sempre que uma consulta é marcada é aberto um prontuário automatizado do cliente que conterá os históricos das consultas já realizadas pelo paciente na clínica. A marcação de consultas para os pacientes está relacionada à agenda do médico. Esta agenda deve ser emitida diariamente para controle das consultas do médico. Diferentes tipos de exames também podem ser marcados e realizados na Clínica. Cada tipo de exame requer determinadas condições do paciente, como jejum de 24 hs e vasilhames específicos. Os históricos de todos os exames marcados devem ficar armazenados no sistema. O resultado final do exame deverá estar automatizado contendo a conclusão final do exame. Atualmente, a clínica atende também através de convênios, que cobrem consultas e exames. O paciente só pode estar vinculado a um convênio. É importante saber quais os tipos de exames da clínica uma determinada empresa conveniada cobre. Um paciente pode marcar, cancelar e remarcar consultas e exames. Isto determina o status da consulta ou do exame. A fatura aos conveniados é emitida ao final do mês e para consultas estratégicas os dados das faturas devem ficar armazenados no banco de dados do sistema. Diversas consultas podem ainda ser realizadas como médicos de uma dada especialidade com horários disponíveis para um dado dia e hora, dias e horários de atendimentos de um dado médico, horários de consultas e exames de um dado paciente, especialidade mais procurada, faturamento no mês com consultas e exames e conveniadas em débitos com a clínica.

8. Sistema para uma escolinha de esportes

Um clube é freqüentado por sócios titulares e seus dependentes. Quando um sócio se associa ao clube é gerado uma carteirinha para ele e seus dependentes que podem ser incluídos depois. Para freqüentar as dependências do clube todo mês o sócio titular deve pagar uma taxa que varia de acordo com a quantidade de dependentes que ele possui. Para controlar o pagamento desta taxa o sistema deve gerar uma boleta contendo os dados do sócio titular, mês de pagamento, valor total, data de vencimento, data de pagamento e multa. Estes dados devem ser armazenados no banco de dados. O clube oferece várias aulas de um determinado esporte. A cada semestre são montadas várias turmas em diversos horários para que o sócio (titular ou dependente) possa se matricular. Dessa forma os sócios e seus dependentes terão escolha na hora da matrícula. Mensalmente deve ser emitido um relatório de oferta de turmas por esporte. O controle dos professores e as turmas de esportes que lecionam deve ser feito pelo sistema. Deve ser gerado um relatório geral de sócios por turmas com os respectivos professores. A folha de pagamento e contra cheque dos professores também é controlada pelo sistema de acordo com as turmas pelas quais são responsáveis. Todo mês é emitido uma fatura ao sócio titular contendo dados das turmas, valor total, data de vencimento e data de pagamento, de acordo com as aulas que ele e seus sócios dependentes fizeram no mês. Deve ser mantido no sistema todos os históricos das faturas pagas, das matrículas dos alunos e dos esportes já lecionados pelos professores e controle de vagas disponíveis. Para efeito da legislação todos os históricos de contra cheques pagos aos professores devem ser mantidos no sistema. A emissão de boletas, faturas, carteirinhas e contra cheques pode ser feito a qualquer momento pelo sistema.

3. Modelo Relacional

- Segunda etapa do projeto de Banco de Dados para um SI, cujo objetivo é obter uma descrição de como implementar o BD dependente do hardware e software escolhido.
- Existem vários tipos de SGBDs no mercado que implementam diferentes modelos: Modelo Relacional, Modelo de Redes, Modelo Hierárquico, Modelo Orientado a Objetos, etc.

3.1 MODELO RELACIONAL DE BANCO DE DADOS

- Proposto por Edgar F. Codd nos anos 70 dando origem aos SGBDRs.
- Princípio: As linhas de uma tabela representam os fatos descritos nas classes ou entidades.

Tabela DEPARTAMENTO		
Nome	CodDepartamento (PK)	CodFuncGerente (FK)
Engenharia e Produção	1	1
Tecnologia e Informação	2	3
Administração Pessoal	3	2

- Durante a especificação do modelo deve-se destacar o tipo, o tamanho e o domínio do atributo que está sendo especificado de acordo com o SGBDR.

Coluna	Tipo	Tamanho	Domínio
Nome	Char	30	a-z, A-Z
CodDepartamento	Int	3	0 a 999
CodFuncGerente	Int	4	0 a 9999

- Principais tipos de dados do SGBD Microsoft SQL Server: Tinyint, Smallint, Int, Bigint, Decimal (n,m), Real, Float, Char(n), Varchar(n), Text, Datetime, Smalldatetime, Money, Smallmoney, Bit, Image.
- Toda tabela deve possuir a **chave primária** (Primary Key - PK) que deve ser mínima.
- As ***Associações/Relacionamentos*** entre duas classes/entidades do modelo conceitual são implementados no modelo relacional com o uso de **chaves estrangeiras** (Foreign Key - FK).
- A FK é formada por um ou mais atributos chaves da tabela original que são adicionados numa das tabelas relacionadas, propiciando uma ligação lógica entre as linhas das tabelas.

Tabela DEPARTAMENTO		
Nome	CodDepartamento (PK)	CodFuncGerente (FK)
Engenharia e Produção	1	1
Tecnologia e Informação	2	3
Administração Pessoal	3	2

Tabela FUNCIONARIO				
Nome	CodFunc (PK)	CodDepartamento (FK)	CodFuncSupervisor (FK)	Salário
João Luiz	1	2	NULL	3.000,00
Fernanda	2	1	1	2.500,00
Ricardo	3	2	1	2.300,00
Renata	4	NULL	2	4.200,00

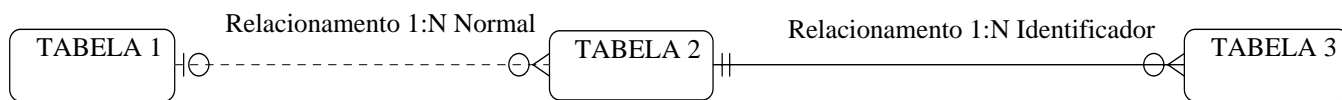
- As FKs não precisam ter o mesmo nome das chaves primárias.

- Os atributos FKs precisam ter do mesmo tipo de dado que as PKs que referenciam.
- A abordagem relacional gera a restrição de **integridade referencial**:
Os valores possíveis das FKs devem pertencer ao conjunto das PKs referenciadas e existentes atualmente no BD ou possivelmente serem NULL. Não pode existir na chave estrangeira um valor que não exista na tabela na qual ela é chave primária.
- A integridade referencial cria problemas durante a manipulação (atualização e remoção) de linhas nas tabelas: Exemplos:
 - a) Como remover o depto 3?
 - b) Como atualizar o Código do departamento 2 para o valor 4?
- Soluções possíveis para os problemas de atualização da PK ou remoção de linhas que sejam referenciadas com FKs em outras tabelas:
 1. **Bloqueia (Restrict/No Action)**: não atualiza/remove enquanto as linhas relacionadas existirem.
 2. **Propaga (Cascade)**: Remove/Atualiza as linhas relacionadas.
 3. **Preenche NULL (Set Null)**: Coloca NULL na coluna que for FK, e remove/altera a PK.
 4. **Preenche o valor Padrão (Set Default)**: Coloca o valor padrão do atributo FK e remove/altera a PK
- Os SGBDRs implementam estas soluções. O analista com base na realidade que está implementando deve decidir qual é a melhor solução para cada um dos relacionamentos e documentá-la adequadamente.
- No caso do software CASE Microsoft Visio estaremos utilizando a seguinte notação de representação das opções de relacionamento:
u: R, C, N ou D *d: R, C, N ou D.*

3.2 MAPEAMENTO DO MODELO CONCEITUAL PARA O MODELO RELACIONAL

Regras de Transformação:

- Durante a aplicação das regras levar em consideração o desempenho, a manutenção e a perda de espaço de armazenamento nesta ordem.
- Para representar o DER Físico Relacional só precisamos dos seguintes símbolos:



Regra 1: Cada classe/entidade forte vira uma tabela, cada atributo simples vira uma coluna com um tipo de dado e domínio a ser definido. Identificar a chave primária (PK) e representá-la com a palavra PK na frente do nome do atributo.

Regra 2: Associações/Relacionamentos 1:N: Acrescente o atributo chave do lado 1 na tabela do lado N como FK contendo um relacionamento 1:N não identificador. Os atributos dos relacionamentos serão acrescentados na tabela do lado N. Se houver a participação parcial do lado 1 a chave estrangeira pode ser NULL. Identifique o atributo que é chave estrangeira com a palavra FK. Se a tabela tiver mais de uma FK identifique-as com as palavras FK1, FK2, etc.

Exercício:

1) Relacionamento 1:N com participação total e tendo atributos próprios:

Departamento(Sigla, Nome)

Empregado (CodEmpregado, Nome, Endereco, ...)

Lotacao [1:N][Departamento, Empregado] (DataInicio)

2) Relacionamento 1:N com participação parcial e tendo atributos próprios:

Departamento(Sigla, Nome)

Empregado (CodEmpregado, Nome, Endereco, ...)

Lotacao [1:N][Departamento, Empregado] (DataInicio)

Regra 3: Toda associação/relacionamento N:N vira uma tabela contendo dois relacionamentos identificadores. As chaves primárias das tabelas relacionadas viram PK e FK na nova tabela gerada. Os atributos dos relacionamentos se existirem ficam na tabela gerada. As FKs não podem aceitar valores NULL. Estes atributos serão identificados com as palavras PK,FK na frente do nome do atributo que tiver esta característica.

Exercícios: Aplique a regra aos seguintes casos:

1) Pedido(NrPedido, ValorTotal, Data)

Produto(CodProduto, Descricao, QuantidadeEstoque, PrecoVenda)

Contém[N:N][Pedido, Produto](Quantidade, PreçoPago)

2) Medico(CRM, Nome, Endereco)

Paciente(CodPaciente, Nome, Endereco)

Consulta[N:N][Medico, Paciente](Data, ValorPago, Diagnostico)

Regra 4: Associações/Relacionamentos 1:1. Adicione a PK de uma das tabelas como FK na outra tabela. Represente o relacionamento como 1:1 não identificador. Os atributos do relacionamento serão colocados na tabela que possuir a FK. Use a seguinte estratégia:

- a) Se houver participação parcial/total ou total/parcial coloque a FK na tabela que tem participação parcial (por que?).
- b) Caso contrário coloque, se possível, a FK na tabela que possua menos linhas.

Exercício: Participação total/parcial
Professor(CodProf, Nome, Endereco)
Curso(CodCurso, Descricao)
Coordena[1:1][Professor, Curso](Gratificacao)

Regra 5: Composição da UML/Entidade Fraca: A classe/entidade fraca vira uma nova tabela. Adicione a PK da tabela do lado 1 como PF,FK na tabela do lado N. Escolha um ou mais atributos da tabela do lado N para compor a PK. Represente o relacionamento como 1:N identificador. O atributos dos relacionamentos serão acrescentados na tabela do lado N. Inclua a opção Cascade para a opção de implementação relacionamento na remoção (d:C).

Exercício: Funcionario (Nome, Endereco, ...)
Dependente (NrDependente, Nome)
Possui[1:N - Composição][Funcionario, Dependente]

Regra 6: AutoRelacionamentos: Aplique as regras para relacionamentos 1:N, N:N e 1:1.

Exercício: Disciplina(CodDisciplina, Nome, CargaHoraria)
Pre-Requisito [N:N][Disciplina, Disciplina].

Regra 7: Associações/Relacionamentos de grau superior a dois: Sempre gere uma tabela tendo como PF,FK as PKs originais das tabelas relacionadas.

Exercício: Voto de legenda do sistema de eleição.

Regra 8: Herança: As seguintes soluções podem ser adotadas:

- a) Solução Padrão: gere uma tabela para a classe/entidade pai e para cada classe/entidade filha. Propague a PK da tabela Pai com PK,FK nas tabelas filhas utilizando o relacionamento 1:1 identificador.
- b) Pode-se gerar uma única tabela acrescentando todos os atributos das classes/entidades filhas na tabela gerada. Crie vários atributos "flags" que identifiquem cada uma das entidades filhas. Esta opção pode gerar um grande número de valores nulos e não é boa se as classes/entidades filhas participam de algum relacionamento.
- c) Outra opção seria gerar duas tabelas para cada classe/entidade filha. Os atributos da classe pai são propagados para as tabelas geradas. Esta opção não é boa se a classe/entidade pai se relaciona com outras classes/entidades.

Exercício: a) Pessoa, Pessoa Física e Pessoa Jurídica.
b) Funcionario, Engenheiro, Administrador.

Exercícios: Desenhe o modelo conceitual e em seguida proponha o modelo relacional para os seguintes casos:

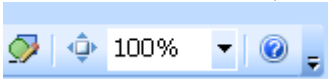
- a) Filme (CodFilme, Nome)
Ator (CodAtor, Nome, Sexo)
Elenco [N:N][Filme, Ator]
- b) Estado (Sigla, NomeEstado)
Governador (CodGovernador, NomeGovernador)
Governa [1:1][Estado, Governador] (DataInicio)
- c) Cliente (Nome)
ClienteTitular (CPF)
ClienteDependente (DataNascimento)
[Herança][Cliente, ClienteTitular, ClienteDependente]
Possui [1:N][ClienteTitular, ClienteDependente]
- e) Pessoa (Nome)
Mãe [1:N][Pessoa, Pessoa]
Pai [1:N][Pessoa, Pessoa]
- f) Prédio (Endereço)
Sala (Numero)
Tem [1:N - Composição][Prédio, Sala]
- g) Aluno (RA, Nome)
Disciplina (Nome)
TurmaDisciplina (Vagas)
Matricula [N:N][Aluno, TurmaDisciplina] (Nota, Frequencia)
Tem [1:N - Composição][Disciplina, TurmaDisciplina]
- h) Cliente (Nome), Filme (Titulo), Exemplar (NrExemplar, DataCompraExemplar),
Ator (Nome), Gênero (Nome), Parâmetros (NomeLocadora, ValorDiariaMulta),
TemExemplar [1:N - Composição][Filme, Exemplar],
Empréstimo [N:N][Cliente, Exemplar] (DataEmprestimo, ValorPago)
AtorFilme [N:N][Ator, Filme], TemGenero [1:N][Gênero, Filme], Édependente [1:N][Cliente, Cliente]
- i) Usuario (nome), Pedido (NrPedido, Descricao), StatusPedido (Nome),
FazPedido [N:N][Usuario, Pedido] (Data),
TemStatus [1:N][StatusPedido, Pedido]
- j) Socio (Nome), Esporte (Nome), TurmaEsporte(Vagas), Categoria(Descricao),
Tem[1:N – Composicao][Esporte, TurmaEsporte],
FazTurmaEsporte[N:N][Socio, TurmaEsporte] (ValorPago),
Possui[1:N][Categoria, Esporte]
- k) Cliente (Nome), Solicitacao (NrSolicitacao, Descricao), StatusSolicitacao (Descricao),
FazSolicitacao [N:N][Cliente, Solicitacao] (Data),
TemStatus [1:N][StatusSolicitacao, Solicitacao]

Como configurar o software Visio para desenhar o DER Lógico (Relacional)

Objetivo: Aprender a usar e configurar o software Visio para construir o Diagrama de Entidade Relacionamento (Modelo Relacional de Banco de Dados).

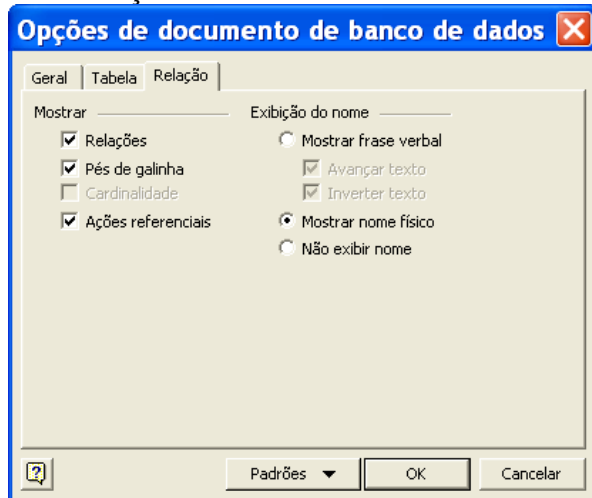
Siga as instruções abaixo:

1. Inicie o Software Microsoft Visio.
2. Selecione a opção “Software e Banco de Dados” (Canto inferior esquerdo).
3. Selecione a opção “Diagrama de Modelo de Banco de Dados”.
4. Aumente o campo de visualização do ZOOM para no mínimo 80% (canto superior direito da sua tela conforme tela abaixo).

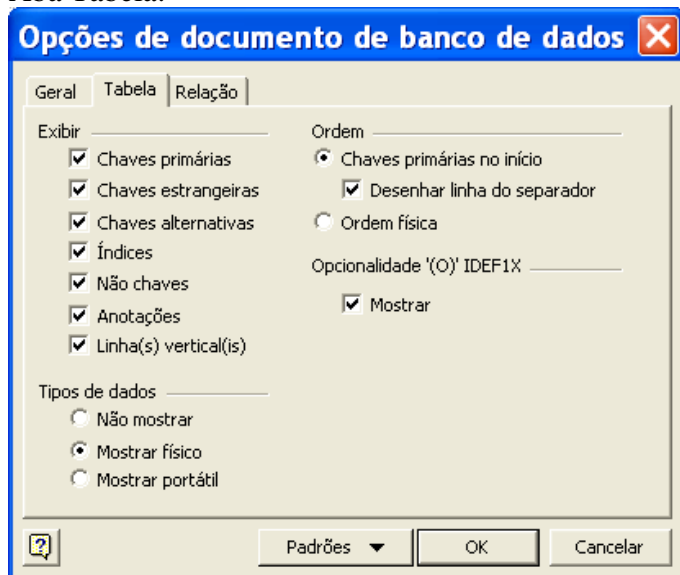


5. Vá na opção de Menu (canto superior) e escolha “Banco de Dados” + “Opções” + “Drivers” + “Microsoft SQL Server”. Você está escolhendo para qual SGBD será desenhado o DER. Observe que existem opções para Oracle, DB2, Access, etc.
6. Novamente na opção de Menu, escolha “Banco de Dados” + “Opções” + “Documento” + escolha as opções para cada aba de acordo com os desenho abaixo:

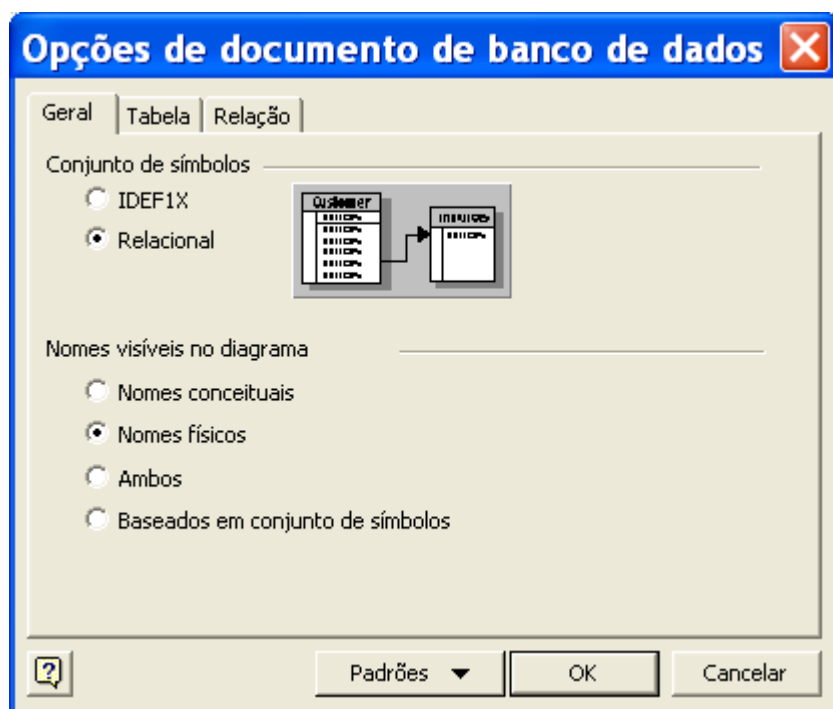
Aba Relação:



Aba Tabela:



Aba Geral:



Clique em OK e comece a desenhar o DER solicitado pelo professor usando os símbolos de Entidade e Relação.

3.3 NORMALIZAÇÃO

- Processo que procura dividir uma tabela em várias outras evitando-se redundância de dados e anomalias de atualização de atributos de tabelas.
- Um bom projeto conceitual de BD (DCP) quase sempre evita o processo de normalização.
- O processo de normalização é baseado em três formas normais principais (existem outras, mas as três primeiras são suficientes para a maioria dos casos).

Primeira Forma Normal (1FN): Uma tabela não pode conter atributos multivalorados. Em outras palavras uma tabela não pode conter grupos de atributos repetitivos.

Exercícios: Coloque na 1 FN as seguintes tabelas.

- 1) Empregado(CodFunc PK, Nome, Telefone[0..*])
- 2) Pedido (NrPedido PK, PrazoEntrega, NomeCliente, EnderecoCliente, CodProduto, QuantidadeProdutoComprado, NomeProduto, PrecoAtualProduto, ValorTotalPedido, CodVendedor, NomeVendedor)

Segunda Forma Normal (2FN): Uma tabela está na 2FN se estiver na 1FN e se ela não possuir atributos que dependem parcialmente da chave primária composta. Ou seja, a tabela não pode conter atributos com dependência parcial da chave.

Obs: Se uma tabela possuir só um atributo chave e estiver na 1FN então automaticamente estará na 2FN.

Terceira Forma Normal (3FN): Uma tabela está na 3FN se estiver na 2FN e se ela não possuir atributos não chaves que dependem de outro atributo não chave. Ou seja, a tabela não pode conter atributos com dependência transitiva.

Exercício:

- 1) Qual forma normal as tabelas abaixo não satisfaz? Em seguida normalize-as.
 - a) Boleta(RA PK, DataVencimento PK, NomeAluno, ValorTotal, DataEmissão, DataPagto)
 - b) Aluno(RA PK, CodDisciplina, NomeAluno, NotaDisciplina)
 - c) Pedido(NrPedido PK, DataPedido, NrVendedor, ValorTotalPedido, NomeVendedor)
 - d) HistoricoEscolar(RA PK, CodDisciplina PK, NomeAluno, NomeDisciplina, NotaFinal)
 - e) Pedido(NrPedido PK, Data, CPF, NomeCliente, ValorTotalPedido, CodProduto, NomeProduto)
- 2) Normalize as seguintes tabelas passo a passo:
 - a) Aluno(RA PK, Nome, Endereço, CodSituação, DescriçãoSituação, Telefones, CodDisciplinaMatricula, DescriçãoDisciplina, CargaHorária, Nota, Frequencia, CodCurso, NomeCurso)
 - b) Cliente(CodCliente PK, Nome, Telefone, Endereco, Cidade, PlacaCarroLocado, DataLocacao, ValorFinalLocacao, ModeloCarro, DescricaoModelo, TaxaModelo)
 - c) Diário (CodDisciplina PK, NrTurma PK, Mês, NomeDisciplina, CodProfessor, NomeProf, NrSala, Turno, LocalizaçãoSala, DataEmissão, TotalAulasPrevista, TotalAulasDadas, NrOrdemAluno, RA, NomeAluno, NrFaltas)

- SQL é uma linguagem padrão de alto nível (não procedural) para criar (DDL) e manipular (DML) um banco de dados em SGBDRs e SGBDORs.
- É padronizada pelo ANSI (American National Standards Institute).
- Versões da SQL lançadas: SQL86, SQL89, SQL92, SQL99 e SQL2003.
- Para efeitos didáticos e práticos de SQL vamos trabalhar com o seguinte estudo de caso:

[illegible]

1. O atributo DataInicioCargo foi colocado na tabela Funcionário (regra 2 do mapeamento conceitual/relacional).
2. Observe a necessidade do atributo DataInicioLotacao para compor a chave primária da tabela Lotação.
3. Em todos os relacionamentos foram definidos as opções de implementações dos mesmos: u: R,C ou N e d: R,C ou N.
4. Observe que somente um atributo é chave primária das tabelas Engenheiro e Administrador (regra 8 do mapeamento conceitual/relacional).
5. Observe que os relacionamentos FE e FA são 1:1 identificadores.
6. Observe a coerência com as opções de implementações dos relacionamentos FE e FA. Ambos são Cascade para a remoção.
7. O atributo CodFuncSupervisor foi colocado na tabela Funcionário (regra 2 do mapeamento conceitual/relacional). No Microsoft Visio é necessário criar o atributo manualmente e criar o relacionamento também manualmente.
8. Todas as chaves estrangeiras estão indexadas. I1, I2 e I3 são índices que foram criados nos respectivos atributos. Estudaremos índices em Banco de Dados II.
9. Responda: o que acontece com as correspondentes linhas nas outras tabelas se um funcionário da empresa for removido?

4.1 DEFINIÇÃO E MANIPULAÇÃO DE DADOS EM SQL

- Criando/Removendo o Banco de Dados:

```
CREATE DATABASE pessoal
```

```
DROP DATABASE pessoal
```

- Criando tabelas:

```
CREATE TABLE Cargo
```

```
(  CodCargo      INTEGER      NOT NULL,  
   Descricao     VARCHAR(30)  NOT NULL,  
   SalarioCargo  MONEY        NULL  
)
```

```
ALTER TABLE Cargo ADD PRIMARY KEY(CodCargo)
```

```
CREATE TABLE TrabalhaEm
```

```
(  
   CodFunc        INTEGER      NOT NULL,  
   CodProjeto     INTEGER      NOT NULL,  
   Total Horas    DECIMAL(3,1) NOT NULL  
)
```

```
ALTER TABLE TrabalhaEm ADD PRIMARY KEY (CodFunc, CodProjeto)
```

```
ALTER TABLE TrabalhaEm ADD FOREIGN KEY (CodFunc) REFERENCES  
Funcionario (CodFunc) ON DELETE Restrict ON UPDATE Cascade
```

```
ALTER TABLE TrabalhaEm ADD FOREIGN KEY(CodProjeto) REFERENCES  
Projeto (CodProjeto) ON DELETE Restrict ON UPDATE Cascade
```

```
CREATE TABLE Funcionario (
```

```
   CodFunc        INTEGER NOT NULL,  
   Nome           CHAR(50) NOT NULL,  
   Salario        MONEY   NULL CHECK (Salario >= 0 OR Salario IS NULL),  
   CodCargo       INTEGER DEFAULT (1),  
   DataNascimento DATETIME,  
   PRIMARY KEY (CodFunc)
```

```
)
```

- Alterando a estrutura de uma tabela:

```
ALTER TABLE Dependentes ADD telefone char(15) NULL.
```

```
ALTER TABLE Dependentes ALTER COLUMN telefone char(20) NULL
```

```
ALTER TABLE Dependentes DROP COLUMN telefone.
```

- Removendo tabela:

```
DROP TABLE Funcionario
```

```
DROP TABLE Cargo
```

- ***Manipulando Linhas nas Tabelas:***

Para inserir, alterar e excluir linhas em tabelas deve-se satisfazer as restrições de chaves (PKs), de integridades referenciais (FKs) e as restrições de domínio.

- Exemplos:

```
INSERT INTO Cargo (CodCargo, Descricao, SalarioCargo)
VALUES (1,'Engenheiro', 1890.00)
```

```
INSERT INTO Cargo VALUES (2,'Administrador', 2780.00)
```

```
INSERT INTO Lotacao (CodFunc, CodDepartamento, DataInicioLotacao, DataFimLotacao)
VALUES (1, 2, '01/04/2000', NULL )
```

Obs: O funcionário 1 e o Departamento 2 já devem ter sido inseridos antes.

```
DELETE FROM Funcionario
WHERE CodFunc = 1
```

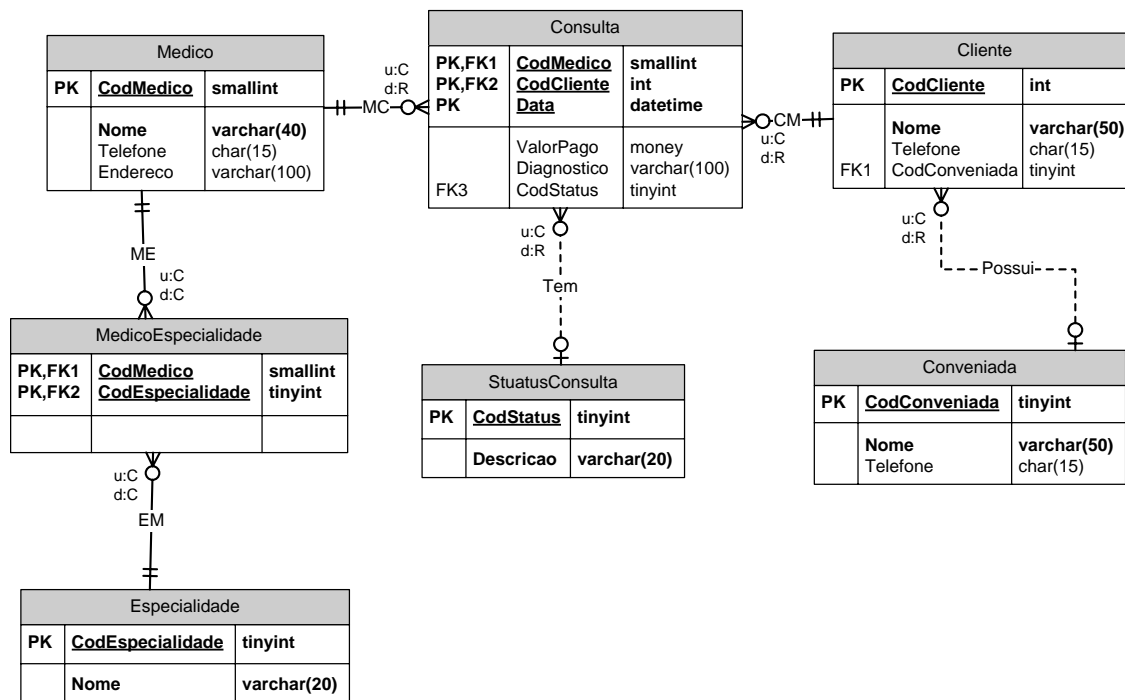
```
DELETE FROM Funcionario
WHERE Salario > 200
```

```
DELETE FROM Funcionario
```

```
UPDATE Funcionario
SET Salário = 1000
WHERE CodFunc = 10
```

```
UPDATE Funcionario
SET Salario = Salario * 1.20
```


Exercício: Observe o DER Relacional abaixo para um sistema de uma clínica médica.



- 1) Com base no DER relacional acima proponha o script SQL padrão ANSI de criação do Banco de Dados.
- 2) Proponha o código SQL que insira duas linhas em cada tabela do BD na ordem correta.
- 3) Proponha o código SQL que remova uma linha da tabela Consulta.
- 4) Proponha o código SQL que remova todas as consultas com status igual a 5.
- 5) Proponha o código SQL que atualize o endereço do médico de código 3.
- 6) Proponha o DCP a partir do DER relacional acima.

4.2 CONSULTAS EM SQL

- Uma consulta em SQL pode ter a seguinte forma:

```
SELECT atributo(s)
FROM tabela(s)
[WHERE condição de seleção]
[GROUP BY atributo(s)]
[HAVING condição de pesquisa]
[ORDER BY atributo(s)]
```

Exemplo 1: Recupere todos os Departamentos da Empresa:

a) Opção 1:

```
SELECT CodDepartamento, Nome, CodFuncGerente, DataInicioGerencia
FROM Departamento
```

b) Opção 2:

```
SELECT *
FROM Departamento
```

Exemplo 2: Recupere o Nome e Endereço dos Funcionários que tenham salário igual ou inferior a 500.00 reais e que tenham sido admitidos após "31/12/2000".

```
SELECT Nome, Endereco
FROM Funcionario
WHERE Salario <= 500.00 AND DataAdmissao >= '01/01/2001'
```

Exemplo 3: Recupere os funcionários que tenham supervisores e que foram admitidos no ano de 2000:

```
SELECT * FROM Funcionario
WHERE CodFuncSupervisor IS NOT NULL AND
DataAdmissao BETWEEN '01/01/2000' AND '31/12/2000'
```

Exemplo 4: Recupere os nome e endereço dos Funcionários que residem em BH e que sejam supervisionados pelos funcionários de códigos 1,4, 6, ou 13:

```
SELECT Nome, Endereco
FROM Funcionario
WHERE Endereco LIKE '%BH%' AND CodFuncSupervisor IN (1,4,6,13)
```

Podem-se usar os seguintes operadores aritméticos (*, /, -, +) nas cláusulas SELECT e WERE.

Exemplo 5: Recupere o nome e Telefone do funcionário que trabalha a mais de um ano na empresa e que teria salário menor que 3000.00 reais caso seu salário atual fosse reajustado em 15 %. Ordene o resultado da consulta pelo Nome do funcionário.

```
SELECT Nome AS "Nome do Funcionário", Telefone AS "Telefone do Funcionário"
FROM Funcionario
WHERE (getdate() - DataAdmissao) > 365 AND (Salario * 1.15) < 3000
ORDER BY Nome
```

4.3 OPERADORES RELACIONAIS

- Várias consultas em um BD podem envolver duas ou mais tabelas do BD.
- A linguagem SQL possibilita a combinação de vários dados de diferentes tabelas em uma única consulta através do JOIN (INNER JOIN).

Exemplo 1: Selecione o nome dos funcionários e o nome dos seus dependentes ordenados pelo nome do funcionário seguido pelo nome do seus dependentes.

```
SELECT    F.Nome AS "Funcionário", D.Nome AS "Dependente"
FROM      Funcionario F INNER JOIN Dependente D ON (F.CodFunc = D.CodFunc)
ORDER BY F.Nome, D.Nome
```

Exemplo 2: Selecione o nome dos departamentos com o nome dos funcionários atualmente lotados nele mostrando o salário de cada funcionário.

```
SELECT    D.Nome, F.Nome, F.Salario
FROM      Departamento D INNER JOIN Lotacao L ON
          D.CodDepartamento=L.CodDepartamento )
          INNER JOIN Funcionario F ON (L.CodFunc = F.CodFunc)
WHERE     L.DataFimLotacao IS NULL
```

Exemplo 3: (Left Outer Join) Recupere o nome dos projetos e o nome de seus coordenadores caso tenha:

```
SELECT    P.Nome AS "Nome do Projeto", F.Nome AS "Nome do Coordenador"
FROM      Projeto P LEFT OUTER JOIN Funcionario F
          ON (P.CodFuncCoordenador = F.CodFunc)
```

Exemplo 4: (Right Outer Join) Mostre o nome de todos funcionários com os respectivos cargos, mesmo para os funcionários que não tenham cargos, ordenados pelo nome do funcionário:

```
SELECT    F.Nome, C.Descricao
FROM      Cargo C RIGHT OUTER JOIN Funcionario F ON (C.CodCargo = F.CodCargo)
ORDER BY F.Nome
```

Exemplo 5: (Full Outer Join) Recupere o nome dos funcionários com seus cargos. Mostre os funcionários que não tenham cargo e os cargos que não tenham funcionários associados:

```
SELECT    F.Nome, C.Descricao
FROM      Funcionario F FULL OUTER JOIN Cargo C ON (F.CodCargo = C.CodCargo)
```

5. Exercícios

5.1 PRÁTICAS DA LINGUAGEM SQL EM LABORATÓRIO

Atenção: Vamos usar o SGBD Microsoft SQL Server para praticar a linguagem SQL padrão ANSI que estamos aprendendo em sala. Para realizar as práticas propostas, você deve **seguir as instruções** abaixo:

Passo 1: Inicialize o SQL SERVER MANAGEMENT STUDIO: Iniciar + Programas + Microsoft SQL Server 2008 + SQL Server Management Studio.

Passo 2: Na janela que irá aparecer informe:

SERVER TYPE: Database Engine

SERVER NAME: srvp1-silvalobo

AUTHENTICATION: SQL Server Authentication

LOGIN: BD

PASSWORD: sql

Clique em OK e será estabelecida a conexão com o Servidor Microsoft SQL Server 2008.

Passo 3: Clique no botão “**New Query**” que se encontra no canto superior esquerda da tela. Será aberta uma nova janela, onde você deverá digitar os comandos SQL que resolvam os problemas propostos. Para executar o comando SQL digitado você deve clicar no botão “**Execute**” no topo da tela.

Atenção: Para realizar as práticas propostas, você deve ter em mãos o DER Relacional do Banco de Dados de Controle Pessoal. Procure sempre responder às questões usando o **padrão ANSI** da linguagem SQL e de forma mais **otimizada** possível.

Prática 1 ***(CREATE TABLE, INSERT, DELETE e UPDATE)***

Considere as seguintes tabelas:

PEDIDO (CODPEDIDO PK, VALORPEDIDO, CODCLIENTE FK)

CLIENTE (CODCLIENTE PK, NOME)

1. Usando SQL, crie as duas tabelas acima com nomes PEDIDO_SEU_RA, CLIENTE_SEU_RA.
2. Usando SQL, adicione as PKs das tabelas do item 1.
3. Usando SQL, adicione a FK na tabela PEDIDO_SEU_RA do item 1.
4. Usando SQL, insira duas linhas em cada uma das tabelas.
5. Usando SQL, atualize o valor de um dos pedidos para 2000 reais.
6. Usando SQL, exclua uma das linhas da tabela CLIENTE_SEU_RA. Atenção, algumas linhas desta tabela não podem ser excluídas (o BD não vai deixar), por quê?

Prática 2 ***(WHERE, Operadores AND, OR, >, <, >=, <=, Order BY ASC/DESC)*** ***(IS NULL, IS NOT NULL, LIKE, IN, BETWEEN, GETDATE)***

1. Recupere o nome e salário de todos os funcionários, ordenados por salário.
2. Recupere todos os pedidos cujo valor total do pedido seja maior que 100 e menor que 1000.

3. Recupere o nome dos produtos cuja quantidade em estoque seja menor ou igual a 10, ordenados pelo preço do produto em ordem decrescente.
4. Recupere o nome e quantidade em estoque dos produtos que tenha quantidade em estoque superior a 5 ou que tenham preço de venda inferior a 100 reais.
5. Recupere o nome dos funcionários que exercem os cargos de código 2 ou 3, ordenados pelo salário destes funcionários em ordem decrescente.
6. Recupere o nome dos dependentes do funcionário de código 2.
7. Recupere todos os pedidos efetuados neste ano, ordenados pela data do pedido.
8. Recupere o nome e salário dos funcionários que não tenham supervisores, ordenados pelo salário. Não devem ser mostrados os empregados demitidos da empresa.
9. Recupere o nome dos dependentes com idade superior a 10 anos. Ordene os dados em pelo nome dos dependentes.
10. Recupere todos pedidos do vendedor de código 2 feitos neste ano.
11. Recupere o nome e endereço dos funcionários que moram na rua “Quebec”, não tenham supervisores, tenham nascidos após o ano de 1980 e ganhem entre 500 e 1000 reais.
12. Recupere o nome, salário e data de nascimento dos funcionários que recebam entre 300 e 2000 reais ou que tenham mais de 30 anos. Ordene os dados em ordem decrescente de salário.
13. Recupere o código do funcionário e o código do departamento de lotação atual dos funcionários de código 1, 3, 4, 8, 9 12 e 15. Use o operador IN.
14. Recupere o nome e data de início dos projetos ainda não encerrados, mostrando primeiro os projetos mais novos.
15. Recupere o nome e salário dos empregados não demitidos, com mais de 20 anos, que não tenham supervisor e não tenham dependentes e que tenham salário entre 500 e 5000 reais. Ordene os dados em ordem decrescente de salário.
16. Recupere o nome e salário atual dos funcionários que tenham 1 ou 2 dependentes e que teriam salário acima de 2000 reais caso o salário atual fossem reajustados em 20 %.
17. Recupere o valor do salário mínimo atual.

Prática 3 ***(Inner Join)***

1. Recupere o nome do funcionário e o nome da formação escolar que ele possui.
2. Recupere o nome dos funcionários não demitidos e o nome dos projetos que estes funcionários trabalham. Ordene os dados pelo nome do funcionário e, em seguida, pelo nome dos projetos que eles trabalham.
3. Recupere o nome e salário dos funcionários não demitidos e o nome do cargo de cada um dos funcionários.
4. Recupere o nome dos funcionários que recebam mais de 500 reais e o nome dos dependentes destes funcionários.
5. Recupere o nome dos funcionários, o nome de seu cargo e o nome de sua formação escolar.
6. Recupere o nome dos projetos, o nome e salário dos funcionários que trabalham no projeto, ordenados pelo nome do projeto e, em seguida, pelo salário do funcionário em ordem decrescente.
7. Recupere o nome do departamento e o nome dos funcionários lotados atualmente nos departamentos.
8. Recupere o nome do departamento e o nome do funcionário gerente do departamento.
9. Recupere o nome do projeto e o nome do funcionário coordenador do projeto.
10. Recupere o nome do departamento e o nome dos funcionários vendedores lotados atualmente no departamento.

Prática 4 ***(Left/Right/Full Outer Join)***

1. Recupere o nome do funcionário e o nome dos projetos que ele trabalha. Mostre todos os funcionários mesmo que ele não trabalhe em nenhum projeto.
2. Recupere o nome todas as formações escolares e o nome de todos os funcionários vinculados a esta formação. Mostre todas as formações e todos os funcionários. Sugestão: utilize o Full outer join.
3. Recupere o nome do departamento e o nome do funcionário gerente. Mostre todos os departamentos.

4. Recupere o nome dos funcionários admitidos a menos de 10 anos e que receba até 2000 reais na empresa. Mostre somente os funcionários não demitidos.
5. Recupere o nome dos projetos e o nome de todos os funcionários que trabalham a mais de 5 horas nestes projetos.
6. Selecione o nome e salário dos funcionários não demitidos, que não tenham dependentes e que moram em “Belo Horizonte”, ordenados pelo salário em ordem decrescente.
7. Selecione o nome de todos os funcionários e o nome do departamento atual de lotação. Mostre todos os funcionários não demitidos.
8. Selecione o nome e salário dos funcionários demitidos há mais de dois anos, ordenados pelo salário.
9. Selecione o nome do projeto e o nome dos funcionários que trabalham no projeto. Mostre todos os projetos, mesmo que não tenha funcionários trabalhando nele.
10. Selecione o nome do departamento e o nome do gerente do departamento. Só mostre os departamentos que tenham gerentes definidos.

5.2 ESTUDOS DE CASOS COMPLEMENTARES

1. Sistema bancário

Um possui diversas agências espalhadas pelo Brasil. Um cliente do banco só pode pertencer a uma destas agências. O cliente solicita abertura de conta corrente e/ou poupança. Existem diversos tipos de movimentações que os clientes podem realizar nas suas contas como saques, depósitos, débitos de cheques, extratos, saldos, etc. Estas movimentações podem ou não serem tarifadas dependendo do tipo do cliente (normal, preferencial, especial, empresarial, etc). Todo dia de vencimento da conta de poupança deve ser executado um processo que atualiza o saldo da conta, baseado numa tabela diária de correção monetária que contém o índice a ser aplicado. Os históricos das correções monetárias aplicadas nas contas poupanças devem ser mantidos pelo sistema. O cliente pode opcionalmente autorizar que determinadas quantias de sua conta corrente sejam depositadas em aplicações financeiras como fundos de ações, CDBs, RDBs, etc. Cada tipo de aplicação financeira possui uma tarifa e prazo da aplicação. Quando do vencimento da aplicação ou quanto solicitado o resgate o sistema automaticamente aplica a taxa do dia para a aplicação. Semanalmente devem ser emitidos diversos relatórios estratégicos que mostrem o desempenho geral de cada agência do banco, clientes com saldos negativos, malas diretas para clientes com saldo acima da média. O cliente pode emitir extratos e saldos de contas (corrente e poupança) e aplicações financeiras em qualquer momento e em qualquer período.

2. Sistema de controle de um congresso de informática

Um congresso consiste na seleção de trabalhos versando cada um sobre um único tema. Para cada tema há um ou mais professores responsáveis pela avaliação dos trabalhos submetidos previamente pelos autores. Pode haver mais de um professor por tema, mas cada professor só pode ser responsável um tema. Não há limite quanto ao número de trabalhos inscritos por cada autor. Cada professor que avalia um trabalho dá uma nota de 0 a 10. Os trabalhos que obtiveram a melhor nota por tema é selecionado originando uma palestra que será apresentada no congresso. As palestras serão realizadas em diversas salas e horários mas cada palestra só será apresentada uma vez. A organização do congresso precisa dispor de informações sobre a quantidades de lugares e os recursos (como retroprojetor, painel, computador) a serem utilizados por cada uma das palestras. A palestra deverá ser apresentada por um dos autores em um tempo previamente combinado para cada um. O público realizará avaliação de cada palestra e a média obtida da palestra deverá ser registrada. Os participantes devem se inscrever para o congresso, podendo assistir qualquer palestra. No certificado de participação deverão constar as palestras assistidas por cada participante. Esta informação será colhida quando da entrada do participante na sala da palestra. É importante manter o cadastro das empresas às quais pertencem os participantes para envio de certificado. Se o participante não estiver vinculado a nenhuma empresa seus dados cadastrais devem ser mantidos pelo sistema. O sistema solicitado deve ser capaz de, informar palestras inscritas por tema, com autor(es) e apresentador, resumo; data, sala e horário de cada palestra; controlar inscrições e frequência dos participantes e emitir os certificados de participação. É importante manter no sistema todas as palestras e trabalhos já apresentados nos congressos anteriores.

3. Sistema para uma agência de turismo

Uma agência de turismo deseja informatizar a oferta e o controle de pacotes de viagens nacionais e internacionais. O gerente usuário forneceu as seguintes informações. Um pacote é um programa turístico nacional ou internacional. Cada pacote turístico envolve várias cidades numa sequência previamente definida pela agência, o que define o roteiro do pacote. O deslocamento para cada cidade do roteiro pode ser feito através de meios de transporte distintos. Em cada cidade envolvida no roteiro o turista pode ficar hospedado em um determinado Hotel conveniado o que compreende a estadia nas cidades do roteiro. Um pacote pode ser realizado várias vezes em épocas diferentes. Cada realização é chamada “oferta” com preço base, período de início e período de término. O turista compra uma oferta de um pacote num determinado período. Cada cidade recebe uma classificação de acordo com seu atrativo principal (ex. estância, praia, histórica, montanha, etc). Uma cidade pode receber mais de uma classificação como a Cidade de Salvador que é histórica e uma cidade praiana. Atrações locais de uma cidade define os pontos turísticos da cidade e podem ser de diversos tipos como museus, restaurantes, praias, igrejas, etc. É importante conhecer a localização dos pontos turísticos das cidades que fazem parte do roteiro, bem como uma breve descrição do ponto turístico e se for pago a visita o preço estimado. Os hotéis conveniados nas cidades que fazem parte do roteiro possuem diferentes categorias (três, quatro ou cinco estrelas). O usuário deseja obter relatórios contendo quais cidades, estados, ou países fazem parte do pacote, quais os hotéis e suas classificações por categoria, quais as atrações turísticas de uma cidade, qual a sequência entre as cidades de um pacote, quais os horários de chegada e saída de cada cidade e qual o meio de transporte utilizado para os deslocamentos entre elas em cada pacote, qual o tempo de permanência em dias em cada cidade, qual o idioma principal de cada país visitado em um pacote internacional, quais os turistas participantes de cada oferta de pacote e de quais pacotes participou o turista, total de vagas em cada oferta, total arrecadado por ofertas de um determinado pacote em um dado mês.

4. Sistema de apuração de uma eleição

A Justiça eleitoral deseja automatizar o processo de votação e apuração dos votos da próxima eleição. Para isso é fornecido o modo de funcionamento da eleição. Cada cidade possui várias zonas eleitorais que funcionam em prédios requisitados pela Justiça Eleitoral, sendo que cada uma das zonas engloba várias seções eleitorais. É importante possuir dados sobre os prédios existentes nas cidades. Várias zonas eleitorais podem funcionar em um mesmo prédio. Em alguns casos, uma zona eleitoral pode ser dividida em mais de um prédio. Um prédio possui várias salas onde funcionam as seções eleitorais. Uma seção eleitoral funciona em somente um prédio. Cada eleitor está vinculado a uma seção eleitoral. Na época da eleição o TRE define as urnas, especifica um número a elas e disponibiliza uma listagem de eleitores para cada urna. Uma seção eleitoral está associada a uma e somente uma urna e vice versa. É convocado pelo TRE com base na formação escolar do eleitor algumas pessoas responsáveis pela urna durante a votação. Isto define a mesa eleitoral composta pelos mesários com títulos como presidente de mesa, primeiro secretário, segundo secretário, etc. Cada candidato

concorre a um cargo (prefeito, deputado municipal, estadual, federal, senador, etc) e é filiado a um único partido político. Para cada cargo disputado o eleitor pode votar no candidato, no partido, em branco ou anular o voto. Os votos são totalizados por urna e o sistema deve emitir os boletins de urna. Deverá produzir uma listagem de eleitores de cada seção eleitoral e o resultado geral da eleição por seção e por zona. Deve ser ainda emitido uma listagem contendo a totalização dos 10 candidatos mais votado por cargo, a quantidade de votos por partido para um dado cargo, a quantidade de votos nulos por cargo e a quantidade de eleitores que não votaram por seção e por zona.

5. Sistema estacionamento Modelo

O estacionamento Modelo funciona como a maioria dos estacionamentos de BH. Ao deixar o carro o cliente recebe um comprovante no qual consta a placa do veículo, a categoria do veículo e a hora da entrada. Ao sair o cliente apresenta o comprovante e paga em função do tempo em que o veículo ficou estacionado e da categoria do veículo. Existem 4 categorias de veículos "Pequeno, Médio, Grande e Especial". Para cada categoria existe uma taxa horária de estacionamento, cobrada a cada 30 minutos. Um recibo de estacionamento com hora de entrada e de saída, placa e categoria do veículo e valor pago é entregue ao cliente após o pagamento da conta. Todos estes dados devem ser armazenados no banco de dados do sistema.

O controle de vagas será automatizado pelo sistema. As vagas do estacionamento são demarcadas em função das categorias, ou seja, dado uma vaga é necessário saber qual a categoria de veículo ela comporta. O recepcionista atribui as vagas desocupadas aos veículos, na ocasião em que eles entram. Por isso o sistema deve exibir um painel mostrando onde se encontra uma vaga livre de acordo com a categoria do veículo. A vaga passa então a ser ocupada e torna-se novamente livre com a saída do veículo. Note que é importante conhecer em que vaga o veículo está estacionado na ocasião da saída, desta maneira o manobrista não perde tempo em procurá-lo. Além dos clientes avulsos, o estacionamento Modelo atende a mensalistas e empresas conveniadas. Os mensalistas pagam uma boleto mensal de acordo com a categoria dos veículos que possuem e têm direito a estacionar o carro a qualquer hora do dia. Ao entrar no estacionamento, eles identificam-se como mensalistas por meio de uma carteirinha especial, emitida na ocasião do pagamento da mensalidade. Caso um mensalista inicie um contrato durante o mês o valor é calculado de maneira proporcional considerando-se os dias que faltam até o final do mês (mês padrão de 30 dias). Na carteirinha de mensalistas consta o mês de vigência e dados do mensalista. É importante saber quais os veículos um mensalista possui. As empresas conveniadas fazem um contrato com o estacionamento Modelo no sentido de evitar que seus visitantes paguem o estacionamento. Dados da empresa devem ser armazenados como o número, CGC da empresa, endereço, data do contrato, etc. No caso das conveniadas o estacionamento Modelo emite cupons contendo um número único, o nome da empresa e dados do estacionamento e os envia às empresas conveniadas quando solicitado. É importante manter o controle dos cupons emitidos. O funcionário responsável na empresa conveniada deve carimbar os cupons relativos aos veículos dos visitantes num espaço especialmente reservado para essa finalidade. Neste caso o estacionamento não emite um recibo, mas envia ao final de cada mês uma fatura aos conveniados constando a relação dos números dos cupons utilizados pela empresa. Sobre o total da fatura é aplicado um desconto padrão de 15%. Para efeitos legais o estacionamento deve manter todos os históricos de ocupações de veículos junto às vagas do estacionamento contendo a placa do veículo, a vaga ocupada, a hora início e a hora fim que o veículo ocupou a vaga. Não é importante guardar dados dos clientes avulsos. O controle de status das boletas mensais, faturas e cupons deve ser feito pelo sistema. A qualquer momento pode ser gerado relatórios contendo a taxa de ocupação das vagas, valores recebidos com diaristas, conveniados e mensalistas

6. Sistema para controle de cinemas

Deseja-se projetar um sistema que atenda às necessidades de controle dos cinemas e filmes de uma determinada empresa de distribuição de filmes no país. A empresa de distribuição possui diversos escritórios espalhados pelos municípios do país. Os escritórios são responsáveis por distribuir os filmes para os diversos cinemas da cidade. Em cada município existe no máximo um escritório de distribuição mas um município pode possuir vários cinemas. Cada cinema possui uma identificação única, um nome fantasia, um endereço completo, incluindo rua, avenida, bairro e CEP. Um cinema pode possuir várias salas de exibição e é necessário conhecer a capacidade de lotação destas salas. Os filmes são cadastrados no sistema e sempre é classificado em um determinado gênero, o que define a censura do filme. Cada filme é registrado com um título original, ano de produção e sua duração e se for filme estrangeiro possuirá também o país de origem, o título em português e se é legendado ou dublado. Se o filme for nacional é importante saber o nome do principal patrocinador e se foi ou não feito através da lei federal de incentivo à cultura. Informações sobre os atores e atrizes que compõem o elenco principal do filme é importante. Estas informações são nome e a nacionalidade do ator, além disso, saber qual o diretor do filme é interessante. Existirá um único diretor para cada filme. Os atores/atrizes de um filme podem, obviamente, atuar em diversos filmes, assim como o diretor de um filme pode também ser ator neste filme ou ainda mais, ser ator em outro filme. Como já foi dito, alguns cinemas possuem várias salas de exibição e cada sala pode apresentar mais de um filme em cartaz. As sessões possuem horários que variam de acordo com a duração do filme, havendo sempre um intervalo de aproximadamente 15 minutos entre elas. As sessões devem ter seu público registrado diariamente, para que se permita a totalização dos assistentes quando o filme sair de cartaz, ou a qualquer instante. A programação semanal do cinema e suas salas é um relatório importante contendo o dia de exibição do filme, sinopse, diretor, hora início, e uma cotação indicando se o filme é bom ou não. Além das informações acima os usuários necessitam de relatórios como: apuração do público por municípios, por cinemas, e por salas e sessões; permitir uma forma de consulta que dado um determinado ator ou atriz, sejam localizados os cinemas e salas onde estão em cartaz os filmes em que este ator ou atriz atua; em quais cinemas está sendo exibido um determinado gênero de filme; em quais cinemas estão sendo exibido filmes nacionais; e quais filmes de um dado diretor está em cartaz.

7. Sistema hoteleiro

Deseja-se desenvolver um sistema para um hotel que atenda aos seguintes requisitos. O hotel possui diferentes quartos que são classificados em uma das seguintes categorias: Presidencial, Econômica, Simples, Duplo ou Triplo. O valor base da diária dos quartos

depende da categoria do quarto. Quando o cliente telefona ou vem até o hotel e pede para reservar um quarto a recepção verifica se existe quarto disponível no período solicitado. Caso positivo, o cliente é cadastrado e é feita a reserva do quarto. Caso negativo, o cliente é informado da não disponibilidade do quarto. O período possui uma descrição, data início, uma data fim, uma taxa adicional sobre o valor base da diária dos quartos se o período for em alta temporada. Quando o cliente não mais desejar o quarto reservado o funcionário providencia o cancelamento da reserva, disponibilizando novamente o quarto. Quando o cliente não comparecer ao hotel para hospedar-se até às 12:00hs do dia da reserva, essa deve ser cancelada automaticamente pelo sistema. Quando o cliente ocupar um quarto reservado previamente, caracteriza-se uma ocupação do quarto. Neste caso um pacote com informações úteis sobre a utilização dos serviços do hotel é emitido ao cliente. Neste pacote consta o horário do café da manhã, do almoço, das piscinas, os diversos telefones do hotel, pontos turísticos da cidade, etc. Isto tudo deve ser armazenado pelo sistema. Diariamente é verificado se o cliente consumiu algum produto do frigobar do quarto onde está hospedado, para ser armazenado no sistema. No momento de ocupação de um quarto, o cliente recebe uma relação da quantidade e dos preços dos produtos que se encontra no frigobar. Toda vez que um produto é consumido é dado baixa no estoque. Se a quantidade de estoque de um produto estiver baixo é emitido um relatório à gerência solicitando a compra do produto ao fornecedor previamente cadastrado. Quando o cliente deixar o hotel será fornecida a respectiva conta e o quarto estará disponível para limpeza. Quando o quarto estiver limpo a recepção do hotel torna-o disponível. O cliente pode pagar a conta à vista, cheque ou usando cartão de crédito. Dados sobre a conta são importantes serem armazenados como a forma de pagamento, valor da hospedagem, total do frigobar, telefonemas particulares do cliente, o valor total dos serviços adicionais, dia do pagamento e valor total da conta. Diariamente, deve ser emitido um relatório contendo os quartos ocupados e não ocupados e o fluxo de caixa do hotel (contas a pagar e a receber). Malas diretas com as promoções de um período pode ser enviadas aos clientes.

8. Sistema para um grupo de pesquisa sobre vírus

Deseja-se automatizar o processo de publicação de artigos sobre as pesquisas realizadas em um hospital que atenda os seguintes requisitos. Um projeto de pesquisa envolve vários médicos pesquisadores sobre um ou mais tipos de vírus. A informação registrada sobre cada vírus inclui o nome científico e um texto livre para descrição científica do mesmo. O projeto de pesquisa sempre é financiado por instituições exteriores ao hospital sendo possível a participação de médicos externos ao hospital (médicos não residentes). É importante saber a qual instituição externa um médico não residente pertence. Um projeto de pesquisa pode gerar vários artigos que são publicados na edições mensais do jornal científico do hospital, identificado pelo nome do jornal, o número do volume e o número da edição. Um artigo pode abordar vários projetos de pesquisas do hospital. Um artigo pode ter um ou mais médicos autores e ser referente a um ou mais tipos de viroses. O resumo (abstract) do artigo também é armazenado no banco de dados. Um médico residente pode coordenar um ou mais projetos de pesquisas. Um projeto de pesquisa só tem um médico coordenador. Toda vez que um novo projeto de pesquisa é aprovado pela direção do hospital e gerado um contrato de pesquisa contendo o número do contrato, valor financiado, data de início e término. No final de cada mês deve ser gerado vários relatórios como os projetos de pesquisa com seus médicos, relação dos artigos publicados no mês com os respectivos médicos autores, relação dos vírus pesquisados com os projetos nos quais estão envolvidos e instituições financeiras com seus projetos de pesquisa.

9. Sistema de gerência para uma universidade

Decidiu-se automatizar os procedimentos da Gerência Acadêmica (GA) da Universidade UNITESTE. A Gerência Acadêmica mantém um controle centralizado de alunos, cursos, disciplinas, turmas de disciplinas por turno, professores, matrícula e histórico escolar de alunos. Os alunos são admitidos nos cursos através de um vestibular, transferência ou rematrícula. Um aluno só pode estar ligado a um curso, em um dado instante. Os alunos, quando ingressam na universidade, preenchem uma ficha cadastral (com número de matrícula pré-impresso) contendo nome, endereço, telefone, CPF, RG, data de nascimento, semestre de admissão, nome do pai e mãe. De acordo com as normas da UNITESTE cada turma de uma disciplina para ser oferecida necessita de um mínimo de cinco alunos e cada disciplina deverá ter no máximo 55 alunos. A cada semestre são montadas várias turmas em diversos horários para que aluno possa se matricular. Cada horário é definido com base em um código que indica o dia da semana, a hora início e a hora fim. Cada turma montada pode possuir diversos horários e vice versa. Para cada turma montada é definida o professor que vai lecionar e a sala onde vai ser lecionada. Na UNITESTE existem várias unidades acadêmicas (campus), sendo que sala deve pertencer a uma destas unidades acadêmicas. O sistema emite uma boleta que é enviado à casa do aluno de acordo com a quantidade de créditos das turmas de disciplinas que o aluno se matricula no semestre. Os históricos dos pagamentos destas boletas são mantidos no sistema. Os cursos possuem uma grade curricular a qual é composta por diversas disciplinas. Estas podem ser obrigatórias ou optativas dependendo do curso a que pertencem. Cada disciplina está sob a responsabilidade de um departamento da universidade, e a mesma é codificada de acordo com um padrão preestabelecido pelo conselho. Uma disciplina pode ter vários pré-requisitos e ser pré-requisito de diversas outras. A UNITESTE adota um sistema progressivo de aprendizado, no qual cada disciplina pode ter no máximo três e no mínimo zero pré-requisitos. Geralmente, as matérias sem nenhum pré-requisito, ou estão no primeiro período, ou são disciplinas eletivas (não obrigatórias). Segundo uma convenção adotada pela UNITESTE os professores podem ser cadastrados na GA sem estar lecionando uma disciplina. Cada professor pode ministrar até o máximo de 6 turmas. Para que um professor ministre uma disciplina o mesmo deve estar devidamente habilitado pelo CFE (Conselho Federal de Educação). Cada professor está vinculado a um departamento e possui um código específico para sua diferenciação dentre os demais professores. Dados como endereço, telefone, CPF, maior titulação, data de nascimento e valor da hora aula devem ser gerenciados pelo sistema. Quando um professor é contratado é definido no sistema quais as disciplinas cada um deles está habilitado a lecionar. A folha de pagamento e contra cheque dos professores também é controlada pelo sistema de acordo com as turmas pelas quais os professores lecionam no semestre. Dados dos pagamentos devem ser mantidos no sistema. A UNITESTE permite a emissão do histórico escolar do aluno. Este documento é o conjunto de todas as disciplinas cursadas e em curso pelo aluno em toda a sua vida acadêmica dentro da UNITESTE. Contém o registro das disciplinas, e indica a frequência, nota final e o semestre em que a disciplina foi cursada. Os departamentos controlam os cursos de suas áreas de

atuação. As responsabilidades envolvem a definição do número de créditos exigidos para a conclusão do curso, o número total de horas exigidas para o curso e o número total de horas nas disciplinas obrigatórias. Um professor pode coordenar um curso e todo curso deve ter um coordenador. Se o professor coordenar um curso ele possui uma gratificação. A data de início da coordenação é importante ser armazenada pelo sistema. De acordo com o conselho acadêmico um aluno pode, em um dado semestre, não estar matriculado em nenhuma turma de uma disciplina caracterizando um trancamento de matrícula. Em um período letivo, um aluno pode se matricular, no máximo, em sete turmas de disciplinas. O conselho também fixou que um aluno pode repetir no máximo três vezes a mesma disciplina. No total, a UNITESTE pode comportar cinco mil alunos matriculados em seus diversos cursos. A cada ano, são admitidos 800 novos alunos via vestibular e as transferências externas podem ser no máximo 60. Formam-se em torno de 300 alunos por semestre. A UNITESTE oferece atualmente 10 cursos e 280 disciplinas possuindo cerca de 120 professores.

10. Sistema de controle de pessoal de uma empresa

Este sistema tem como objetivo automatizar a área de recursos humanos de uma determinada empresa. Os empregados da empresa são admitidos e ficam lotados em departamentos da empresa. Estes departamentos controlam determinados projetos. É importante saber quais foram todos os departamentos que um empregado já trabalhou. Existem dois tipos especiais de empregados: os empregados que são administradores e os empregados que são engenheiros. Os empregados que são administradores podem gerenciar um departamento e quando isso ocorre eles recebem uma gratificação. Não é importante saber quais foram os gerentes de um dado departamento mas somente quem é o gerente atual. Os empregados que são engenheiros podem trabalhar em projetos previamente criados e é importante saber o dia, a hora e a quantidade de horas, que um empregado trabalhou em um projeto. Todo projeto possui um único empregado coordenador e este coordenador só pode ser um empregado que é do tipo administrador. Se o empregado coordena algum projeto ele também recebe uma gratificação extra. Um empregado pode ser supervisor de vários outros empregados e quem é o supervisor é um fato importante a ser gerenciado no sistema. Um empregado sempre possui um único cargo na empresa, o que define o seu salário base. De acordo com sua formação escolar o empregado pode receber uma gratificação extra. A empresa possui amplo plano de benefícios distribuídos aos funcionários e isto deve ser armazenado no banco de dados. Informações dos dependentes dos funcionários e o tipo de relação entre o funcionário e seu dependente (exemplo: esposa, filho, etc) deve ser controlado no sistema. O contra cheque dos empregados da empresa é gerado no final de cada mês com base no salário fixo e nas participações do empregado nos projetos, gerência ou coordenação de projetos. Os dados do contra cheque emitido ao final de cada mês tais como salário bruto, líquido e descontos devem ser armazenados no sistema para recuperação posterior. Mensalmente devem ser emitidos relatórios estratégicos para o gerente da empresa como o total da folha de pagamento por departamento, o empregos demitidos e admitidos no mês, os projetos em andamento com os respectivos coordenadores, a quantidade de horas trabalhadas por cada empregado em cada projeto, a relação dos benefícios por empregado, a relação de dependentes por empregado, a relação dos gerentes por departamento e a relação dos supervisores por empregado.